Recuperatorio Primer Parcial de Labo I

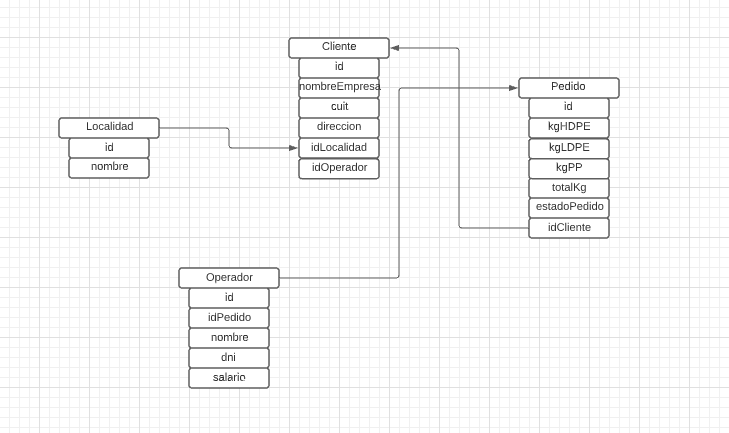
Pedro Sanchez

1F Turno noche

DNI: 96.057.540

[Pjsanchez197@gmail.com](mailto:Pjsanchez197@gmail.com)

Video: https://drive.google.com/file/d/1qxV28eQ6RZGjMiUApDDGOkLReY3X85du/view?usp=sharing



/\*\* \brief Para ingresar una localidad e indicar la cantidad de pedidos pendientes para dicha localidad

\* \param pedidoList Pedido\* Pointer to array of Pedido

\* \param pedidoLen int Pedido array size

\* \param clienteList Cliente\* Pointer to array of Cliente

\* \param clienteLen int Cliente array size

\* \return int Return (-1) if Error [Invalid length or NULL pointer] - (0) if Ok

\*

\*/

**int** **informe\_imprimirLocalidadPendiente**(Pedido \*pedidoList, **int** pedidoLen, Cliente\* clienteList, **int** clienteLen);

/\*\* \brief Muestra la cantidad de kilos de polipropileno reciclado promedio por cliente

\* \param pedidoList Pedido\* Pointer to array of Pedido

\* \param pedidoLen int Pedido array size

\* \param clienteList Cliente\* Pointer to array of Cliente

\* \param clienteLen int Cliente array size

\* \return int Return (-1) if Error [Invalid length or NULL pointer] - (0) if Ok

\*

\*/

**int** **informe\_imprimirPromedioPPCliente**(Pedido \*pedidoList, **int** pedidoLen, Cliente\* clienteList, **int** clienteLen);

/\*\* \brief Se imprimirá una lista de clientes con todos sus datos junto con la cantidad de “pedidos

de recolección” que posee en estado “Pendiente”

\* \param pedidoList Pedido\* Pointer to array of Pedido

\* \param pedidoLen int Pedido array size

\* \param clienteList Cliente\* Pointer to array of Cliente

\* \param clienteLen int Cliente array size

\* \return int Return (-1) if Error [Invalid length or NULL pointer] - (0) if Ok

\*

\*/

**int** **informe\_imprimirClientesConPendientes**(Pedido \*pedidoList, **int** pedidoLen, Cliente\* clienteList, **int** clienteLen);

/\*\* \brief Se imprimirá una lista de los pedidos que se encuentren en estado “Pendiente” con la información: Cuit del cliente, dirección del cliente, cantidad de kilos a recolectar.

\* \param pedidoList Pedido\* Pointer to array of Pedido

\* \param pedidoLen int Pedido array size

\* \param clienteList Cliente\* Pointer to array of Cliente

\* \param clienteLen int Cliente array size

\* \return int Return (-1) if Error [Invalid length or NULL pointer] - (0) if Ok

\*

\*/

**int** **informe\_imprimirPedidosPendientes**(Pedido \*pedidoList, **int** pedidoLen, Cliente\* clienteList, **int** clienteLen);

/\*\* \brief : Se imprimirá una lista de los pedidos que se encuentren en estado “Completado” con la información: Cuit del cliente, dirección del cliente, cantidad de kilos reciclados de cada tipo de plástico

\* \param pedidoList Pedido\* Pointer to array of Pedido

\* \param pedidoLen int Pedido array size

\* \param clienteList Cliente\* Pointer to array of Cliente

\* \param clienteLen int Cliente array size

\* \return int Return (-1) if Error [Invalid length or NULL pointer] - (0) if Ok

\*

\*/

**int** **informe\_imprimirPedidosProcesados**(Pedido \*pedidoList, **int** pedidoLen, Cliente\* clienteList, **int** clienteLen);

**int** **informe\_imprimirMaximo**(Pedido \*pedidoList, **int** pedidoLen, Cliente\* clienteList, **int** clienteLen);

**int** **informe\_imprimirClientesConMasCompletos**(Pedido \*pedidoList, **int** pedidoLen, Cliente\* clienteList, **int** clienteLen);

/\*\*

\* @fn void Operador\_MostrarKGxOperador(Operador, Pedido)

\* @brief Imprime en pantalla todos los campos de la estructura del

\* indice del array recibido

\* @param listaOperador Array donde leer

\* @param \*pedidoList Array donde leer

\*/

**void** **Operador\_MostrarKGxOperador**(Operador listaOperador, Pedido \*pedidoList);

/\*\*

\* @fn int Operador\_MostrarTotalKgXoperador(Cliente\*, int, Pedido\*, int, Operador\*, int)

\* @brief Muestra todos los datos cargados en un array de estructura

\*

\* @param list Array donde lee los datos

\* @param TAM Tamaño del array recibido

\* @param \*pedidoList Array donde lee los datos

\* @param pedidoLen Tamaño del array recibido

\* @param listaOperador Array donde lee los datos

\* @param TAM Tamaño del array recibido

\* @return 0 si fue exitoso, 1 si falló

\*/

**int** **Operador\_MostrarTotalKgXoperador**(Cliente\* list, **int** len, Pedido\* \*pedidoList, **int** pedidoLen, Operador\* listaOperador, **int** TAM);

/\*\* \brief add in a existing list of clients the values received as parameters

\* in the first empty position

\* \param list Cliente\*

\* \param len int

\* \param id int

\* \param name[] char

\* \param lastName[] char

\* \param salary float

\* \param sector int

\* \return int Return (-1) if Error [Invalid length or NULL pointer or without

free space] - (0) if Ok

\*/

//int addCliente(Cliente\* list, int len, int id, char nombreEmpresa[], char cuit[], char direccion[], char localidad[]);

**int** **addCliente**(Cliente\* list, **int** len, **int** id, **char** nombreEmpresa[], **char** cuit[], **char** direccion[], **int** idLocalidad);

/\*\* \brief changes the different parameters of one client

\*

\* \param list Cliente\*

\* \param len int

\* \return ; 0 success, -1 error

\*

\*/

//int ModificarCliente(Cliente\* list, int len);

**int** **ModificarCliente**(Cliente\* list, **int** len, Localidad\* localidadList, **int** localidadLen);

/\*\* \brief Remove a client by Id (put isEmpty Flag in 1)

\*

\* \param list Cliente\*

\* \param len int

\* \param id int

\* \return int Return (-1) if Error [Invalid length or NULL pointer or if can't

find a employee] - (0) if Ok

\*

\*/

**int** **removeCliente**(Cliente\* list, **int** len, **int** id);